AIMMS User's Guide - Getting Started

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit www.aimms.com.

Chapter 2

Getting Started

This chapter provides pointers to the various AIMMS examples that are available online and may help you to get a quick feel for the system. It explains the principle steps involved in creating a new AIMMS project, and it provides you with an overview of the various graphical tools available in AIMMS to help you create a complete Analytic Decision Support application. In addition, the chapter discusses the files related to an AIMMS project.

This chapter

2.1 Getting started with AIMMS

For most people, learning to use a new software tool like AIMMS is made substantially easier by first getting to see a few examples of its use. In this way you can get a quick feel for the AIMMS system, and begin to form a mental picture of its functionality.

Learn by example

In addition, by taking one or more illustrative examples as a starting point, you are able to quickly create simple but meaningful AIMMS projects on your own, without having to read a large amount of documentation. Building such small projects will further enhance your understanding of the AIMMS system, both in its use and in its capabilities.

Getting started quickly

To get you on your way as quickly as possible, the AIMMS system comes with a tutorial consisting of

AIMMS tutorial

- a number of live demos illustrating the basic use of AIMMS,
- an extensive library of small example projects each of which demonstrates a particular component of either the AIMMS language or the end-user interface,
- a number of complete AIMMS applications, and
- worked examples corresponding to chapters in the book on optimization modeling.

The library of small example projects deals with common tasks such as

Example projects

- creating a new project,
- building a model with your project,
- data entry,
- visualizing the results of your model,
- case management, and
- various tools, tips and tricks that help you to increase your productivity.

By quickly browsing through these examples, you will get a good understanding of the paradigms underlying the AIMMS technology, and you will learn the basic steps that are necessary to create a simple, but fully functional modeling application.

What you learn

Rather than providing an introduction to the use of AIMMS, the User's Guide deals, in a linear and fairly detailed fashion, with all relevant aspects of the use of AIMMS and its modeling tools that are necessary to create a complete modeling application. This information enables you to use AIMMS to its full potential, and provides answers to questions that go beyond the scope of the example projects.

This User's Guide

The Language Reference deals with every aspect of AIMMS data types and the modeling language. You may need this information to complete the attribute forms while adding new identifiers to your model, or when you specify the bodies of procedures and functions in your model.

The Language Reference

The Optimization Modeling guide provides you with the basic principles of optimization modeling, and also discusses several advanced optimization techniques which you may find useful when trying to accomplish nontrivial optimization tasks.

The Optimization Modeling guide

The following strategy may help you to use AIMMS as efficiently and quickly as possible.

How to proceed

- Study some of the working examples to get a good feel for the AIMMS system.
- Select an example project that is close to what you wish to achieve, and take it as a starting point for your first modeling project.
- Consult any of the three AIMMS books whenever you need more thorough information about either the use of AIMMS, its language or tips on optimization modeling.

2.2 Creating a new project

Every AIMMS application consists of two main components:

- an AIMMS project file (with a `.aimms` extension), which contains references to the main application project and all library projects contained in your application,
- for the main project and every library project all source files are stored in a separate folder:
    - the `Project.xml` file holding a reference to the project's main model source file (with an `.ams` extension), as well as all additional model source files included in the main model source file, together containing all identifier declarations, procedures and functions that are relevant to the project,
    - `PageManager.xml`, `TemplateManager.xml` and `MenuBuilder.xml` files describing the page, template and menu tree defined in the project, with all individual pages and templates being stored in the `Pages` and `Templates` subfolder,
    - `Settings` and `Tools` subfolders containing the options for the execution engine, and the saved settings for user colors, fonts, and the various tools in the AIMMS IDE, and
    - the `User Files` folder for storing all user files that you store within the project.

Project components

Within an AIMMS session, you can create a new project through the **File-New Project** menu. Note that this menu is only available when no other project is currently open. It will open the AIMMS **New Project** wizard illustrated in Figure 2.1. In this wizard you can enter the name of the new project, along with the directory in which the project is to be stored, and the model file (with the `.ams` extension) to be associated with the project.

Creating a new project

By default, the AIMMS **New Project** wizard suggests that the new project be created in a new subdirectory with the same name as the project itself. You can use the wizard button ![icon] to the right of the location field to modify the location in which the new project is created. However, as AIMMS creates a number of additional files and directories in executing a project, you are strongly advised to store each AIMMS project in a separate directory.

Project directory

By default, the AIMMS **New Project** wizard assumes that you want to create a new model file with the same name as the project file (but with a different extension). You can modify the name suggested by the wizard to another existing or nonexisting model file. If the model associated with the project does not yet exist, it will be automatically created by AIMMS.

Model file
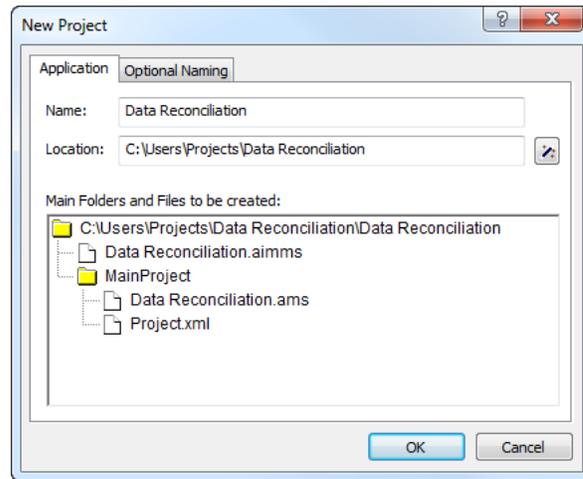
Figure 2.1: The AIMMS **New Project** wizard

After you have finished with the **New Project** wizard, AIMMS will open the *The Model* **Model Explorer**, an example of which is illustrated in Figure 2.2. The Model Ex- *Explorer*
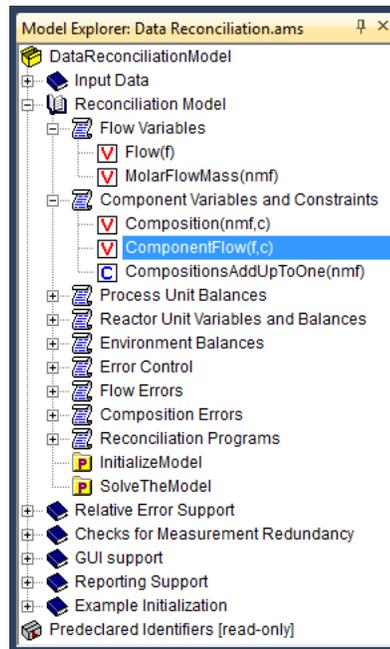
Figure 2.2: The AIMMS Model Explorer

plorer is the main tool in AIMMS to build an AIMMS model, the starting point of building any AIMMS application. In the Model Explorer, the model is presented as a tree of identifier declarations, allowing you to organize your model in a

logical manner and make it easy—both for you and others who have to inspect your model—to find their way around. Besides the Model Explorer, AIMMS provides a number of other development tools for model building, GUI building and data management. An overview of these tools is given in Section 2.3.

You can open an existing AIMMS project in two ways. You can either

Starting an existing project

- start AIMMS and open the project via the **File-Open Project** menu, or
- double click on the AIMMS project file (with a `.aimms` extension) in Windows Explorer.

After opening a project, AIMMS may take further actions (such as automatically opening pages or executing procedures) according to the previously stored project settings.

2.3 Modeling tools

Once you have created a new project and associated a model file with it, AIMMS offers a number of graphical tree-based tools to help you further develop the model and its associated end-user interface. The available tools are:

Modeling tools

- the *Model Explorer*,
- the *Identifier Selector*,
- the *Page Manager*,
- the *Template Manager*, and
- the *Menu Builder* tool.

These tools can be accessed either through the **Tools** menu or via the project toolbar. They are all aimed at reducing the amount of work involved in developing, modifying and maintaining particular aspects of your model-based end-user application. Figure 2.3 provides an overview of the windows associated with each of these tools.

The AIMMS **Model Explorer** provides you with a simple graphical representation of all the identifiers, procedures and functions in your model. All relevant information is stored in the form of a tree, which can be subdivided into named sections to store pieces of similar information in a directory-like structure. The leaf nodes of the tree contain the actual declarations and the procedure and function bodies that make up the core of your modeling application. The **Model Explorer** is discussed in full detail in Chapter 4.

The Model Explorer

While the **Model Explorer** is a very convenient tool to organize all the information in your model, the **Identifier Selector** allows you to select and simultaneously view the attributes of groups of identifiers that share certain functional aspects in your model. By mutual comparison of the important attributes, such overviews may help you to further structure and edit the contents of

The Identifier Selector
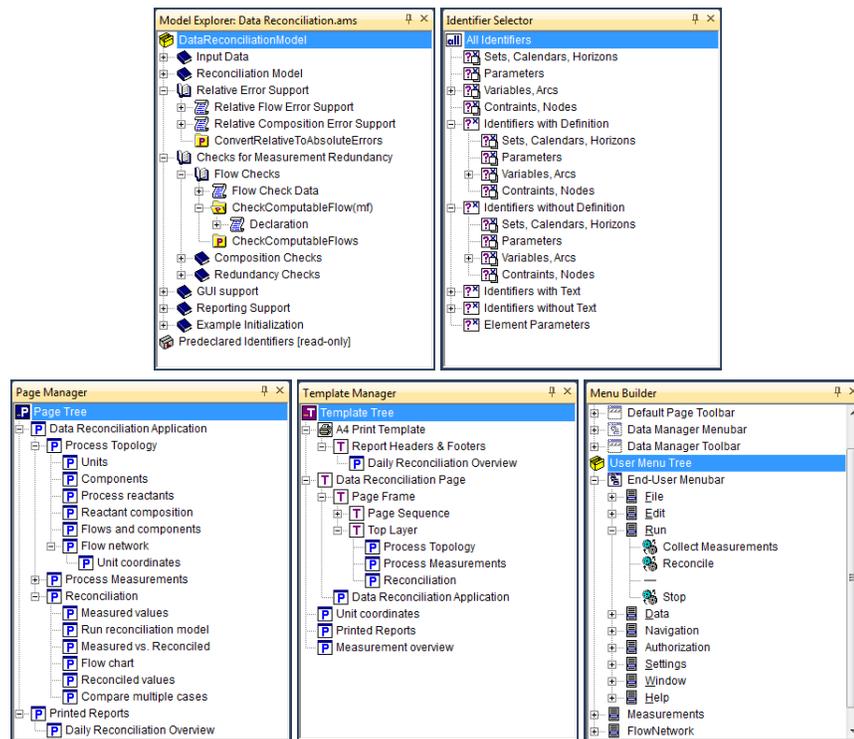
Figure 2.3: Overview of Aimms tools

your model, or to discover oversights in a formulation. The **Identifier Selector** is discussed in full detail in Chapter 7

The **Page Manager** allows you to organize all end-user windows associated with an Aimms application (also referred to as end-user pages) in a tree-like fashion. The organization of pages in the page tree directly defines the navigational structure of the end-user interface. Relative to a particular page in the page tree, the positions of the other pages define common relationships such as *parent* page, *child* page, *next* page or *previous* page, which can used in navigational controls such as buttons and menus. The **Page Manager** is discussed in full detail in Section 12.1.

The Page Manager

Within the **Template Manager,** you can make sure that all end-user pages have the same size and possess the same look and feel. You can accomplish this by creating page templates which define the page properties and objects common to a group of end-user pages, and by subsequently placing all end-user pages into the tree of page templates. The **Template Manager** is discussed in full detail in Section 12.2.

The Template Manager

With the **Menu Builder** you can create customized menu bars, pop-up menus and toolbars that can be linked to either template pages or end-user pages in your application. In the menu builder window you can define menus and toolbars in a tree-like structure similar to the other page-related tools, to indicate the hierarchical ordering of menus, submenus and menu items. The **Menu Builder** is discussed in full detail in Section 12.3.

The Menu Builder

2.4 Dockable windows

Dockable windows are an ideal means to keep frequently used tool windows in an development environment permanently visible, and are a common part of modern Integrated Development Environments (IDE) such as *Visual Studio .NET*.

Support for dockable windows

Dockable windows can be used in a *docked*, *auto-hidden*, or *floating* state. Whether a dockable window is in a docked, auto-hidden or floating state can be changed at runtime through drag-and-drop.

Docking states

When docked, the tool windows are attached to the left, right, top or bottom edge of the client area of the main AIMMS window. By default, all modeling tools discussed in Section 2.3 are docked at the left edge of the AIMMS window, as illustrated in Figure 2.4.

Docked windows
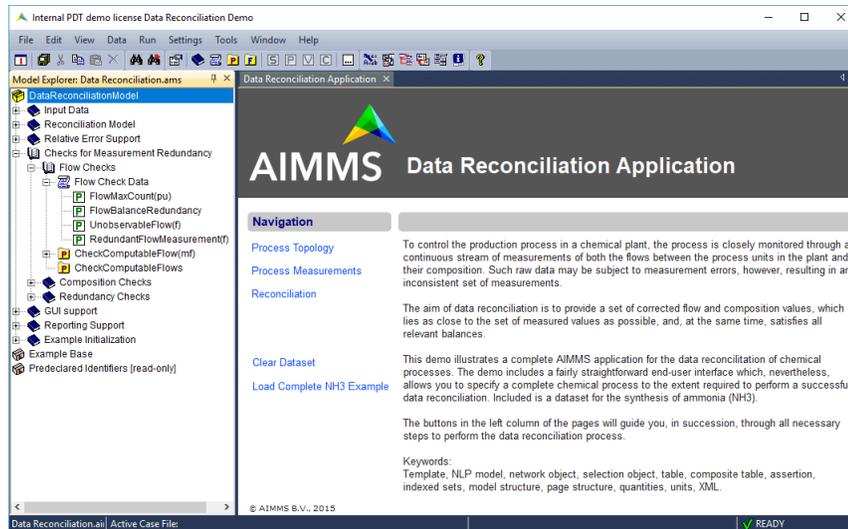
Figure 2.4: Example of a **Model Explorer** docked at the left edge

AIMMS automatically stores the location and size of each tool window. This information is used to restore the location and size of each tool window whenever an AIMMS session is started.

Tool windows persistence

By dragging the windows caption of a docked window and moving the cursor around the edges of the AIMMS window, you can move the docked window to another position. While hovering over a drop target, a blue rectangle (as illustrated in Figure 2.5) snaps into place at the appropriate location, whenever a dockable window is ready to be docked at the location corresponding to the drop target. The area of a docked window can also be split into two by dragging another dockable window into the upper, lower, left or right part of docked window. In all these cases, a blue rectangle shows how a dockable window will be docked when you release the mouse at that time.

Dragging windows
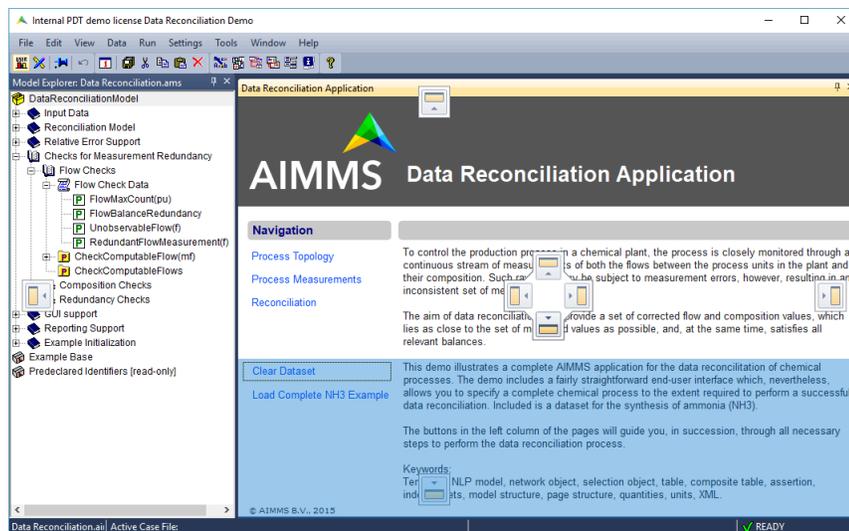
Figure 2.5: Drag-and-drop windows

In auto-hidden state, a dockable window is normally collapsed as a button to the upper, lower, left or right edge of the main AIMMS window. When you push the button associated with a collapsed window, it is expanded. When an expanded tool window looses focus, it is collapsed again. By clicking the push-pin button in the caption of a docked/collapsed window, you can change the window's state from docked to auto-hide and back.

Auto-hidden windows

By dragging a tool window away from an edge of the main AIMMS window, it becomes floating. When AIMMS is the active application, floating tool windows are always visible on top of all other (non-floating) AIMMS windows. Floating windows can also be shown outside the main AIMMS window frame.

Floating tool windows

By default, all pages and attribute forms are shown in *tabbed MDI* mode. In Figure 2.4 the main page of the application is displayed in tabbed MDI mode, with the attribute windows of two identifiers in the model accessible through tabs. Tabbed MDI windows occupy the remaining space of the client area of the main AIMMS window that is not occupied by docked windows. This implies that you do not have control over the size of tabbed MDI windows. Therefore, if you use tabbed MDI mode in your AIMMS application, it makes sense to make all the pages in your model resizable (see Chapter **??**). However, the display mode of a page can be changed to docked (by checking the **Allow User Dockable** option on the **Page Properties** dialog box). This would for example allow you to turn a page into a floating window and display it on you second monitor.

Tabbed MDI mode

When you drag the tab associated with a document window in the document window area, you can move the document window into a new or existing *tab group*, at the left, right, top or bottom of the current tab group. Tab groups effectively split the document window area into multiple parts, each of which can hold one ore more tabbed MDI windows. As with dragging dockable windows, a drag rectangle shows where the window will be positioned if you drop it at that moment. Tab groups are very convenient, for instance, if you want to view two attribute windows simultaneously.

Tab groups

2.5 Additional files related to an AIMMS project

In addition to the AIMMS project folders and files associated discussed in Section 2.2, using an AIMMS project either during development or in a deployment scenario may actually result in the creation of a number of files not mentioned before:

Project-related files

- the name change file (with a `.nch` extension),
- one or more case files (with a `.data` extension),
- a user database file (with a `.usr` extension),
- data backup files (with a `.bak` extension),
- log, error and listing files from both AIMMS and its solvers (with `.log`, `.err`, `.lis` or `.sta` extensions).

AIMMS has the capability to keep track of the name changes you performed on identifiers in your model, and automatically replace an old identifier name by its new one whenever AIMMS encounters a renamed identifier. AIMMS keeps track of the list of such name changes in the name change file (with a `.nch` extension). Each name change is listed in this file on a separate line containing the old name, the new name and the (GMT) time stamp at which the change was actually performed. The automatic name change capabilities of AIMMS are explained in full detail in Section 5.3

Name change file

If you are using a version control system to manage your Aimms sources, it makes sense to also include the name change files under version control. When you change an identifier name, Aimms will not directly refactor all pages to reflect the name change directly, but use the name change file to refactor a page when it is opened. The same is true when opening cases that contain data for the identifier the name of which has been changed. When your changes in a project are merged with another developer's changes, the merged name change file will actually contain all name changes made by both developers.

... and version control

Whenever you save a case in your Aimms project (see also Chapter 16), this will result in the creation of a .data file on disk. By default these case files will be stored in the data subfolder of project's main folder.

Case files

With every end-user project created with Aimms, you can associate an end-user database containing the names and passwords of all end-users entitled to run the project or view its results. Such end-user information is stored in an encrypted format in a user database file (with a .usr extension). You can use a single user database file with more than one project.

The user database

During the execution of your model, all log, error and listing information from both Aimms and its solvers (whether visible in the Aimms **Message** window or not) is copied to log, error and listing files, which, by default, are stored in the Log subdirectory of the project directory. If you are not interested in this information, you can reduce the amount of information that is copied to these log files by modifying the relevant execution options.

Log, error and listing files

Through the **AutoSave & Backups-Data** menu, you can specify that you want Aimms to automatically create backups of the data used during a particular session of your project. The menu will pop up the **Data Backup** dialog box illustrated in Figure 2.6. Similarly as with the project backup files, you can
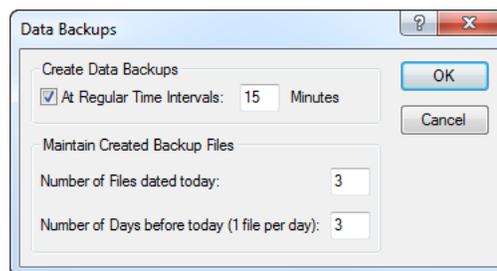
Data backups

Figure 2.6: The **Data Backup** dialog box

indicate whether Aimms should automatically create backup backup files of the session data at regular intervals, as well as how many data backup files

should be retained. Data backup files also have the `.bak` extension and contain a reference to the date/time of the backup.

Besides the automated backup scheme built into AIMMS, you can also create backup files of your session data manually. You can create manual backup files through the **File-Data Backups** menu. When you create a data backup file manually, AIMMS will request a name of a `.bak` file in which the backup is to be stored.

Manually creating backup files

Through the **File-Data Backups** menu, you can restore the data in your application back to the state stored in the data backup files.

Restoring backup files

2.5.1 Project user files

Along with the project-related files created by AIMMS, you may need to distribute some other files with your project when deploying it to your end-users. Such files include, for instance, bitmap files displayed on buttons or in the background of your end-user pages, or files that contain project-related configuration data. Instead of having to include such files as separate files in the project directory, AIMMS also allows you to save them within the project file itself. Both within the AIMMS language as well as in the end-user interface, you can reference such *project user files* as if they were ordinary files on disk.

Project user files

User project files are convenient in a number of situations. The most common reasons to store files as project user files are listed below.

Why use project user files?

■ You want to reduce the number files that you have to ship to your end users. This situation commonly occurs, for instance, when the end-user interface of your project references a large number of bitmap files.
■ You want to hide particular configuration data files from your end-users, which might otherwise only confuse them.
■ User project cannot be modified by your end-users.

You can import files into the project file through the **Tools-Project User Files** menu, which will pop up the **Project User Files** dialog box illustrated in Figure 2.7. In this dialog box, you can create new folders to organize the files you want to import into the project file. The dialog box of Figure 2.7 already contains a folder **bitmaps**, which is automatically added to each new AIMMS project and filled by AIMMS with the bitmaps used on AIMMS' data pages (see Section 5.4). When you are inside a folder (or just within the main project file), you can import a file into it through the **Import File** button, which will open an ordinary file selection dialog box to select the disk file to be imported.
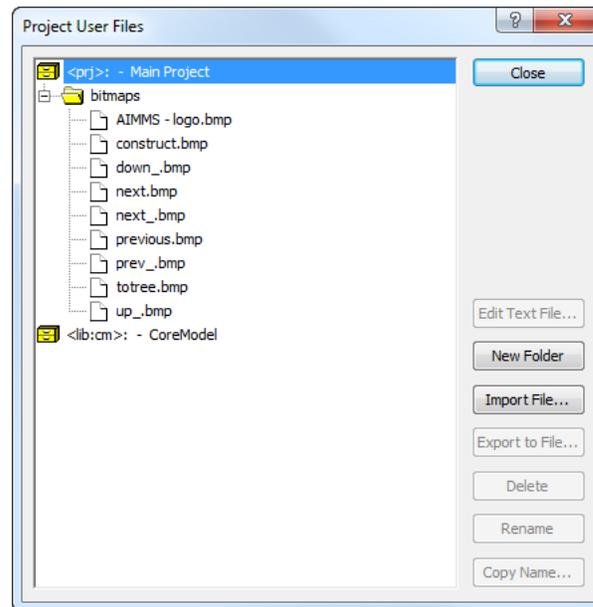
Importing project user files

Figure 2.7: The **Project User Files** dialog box

When your project, next to the main project file, also includes a number of   *User files in*
library project files (see Section 3.1), AIMMS allows you to store user files in   *library projects*
the library project files as well. Thus, if a page defined in a library refers to
a particular bitmap file, you can also store that bitmap as a user file directly
into the corresponding library project file. In the dialog box of Figure 2.7, the
CoreModel node at the root of the tree refers to a library that is included in the
project that serves as the running example throughout this book. Underneath
this node you can add user files that will be stored in the library project file
for the *CoreModel* library.

You can reference project user files both from within the AIMMS language and   *Referencing*
the properties of various objects with the graphical end-user interface. The   *project user files*
basic rule is that AIMMS considers the project file as a virtual disk indicated
by "<prj>". You can use this virtual drive in, for instance, READ, WRITE and PUT
statements within your model. Thus, the statement

```
READ from file "<prj>:config\\english.dat";
```

reads the model data from the project user file "english.dat" contained in a
(developer-created) **config** folder within the project file.

You can access project files in library projects by using the virtual disk no-   *Referencing*
tation "<lib:*library-name*>", where *library-name* is the name of the library   *user files in*
library projects

project. Thus, to read the same file as in the previous paragraph from the *CoreModel* library shown in Figure 2.7, the following statement can be used.

```
READ from file "<lib:CoreModel>:config\\english.dat";
```

Similarly, you can reference project user files on page objects in the end-user interface of your project. Figure 2.8 illustrates the use of a bitmap file stored in the project file on a bitmap button. For all object properties expecting a file

Use in end-user interface
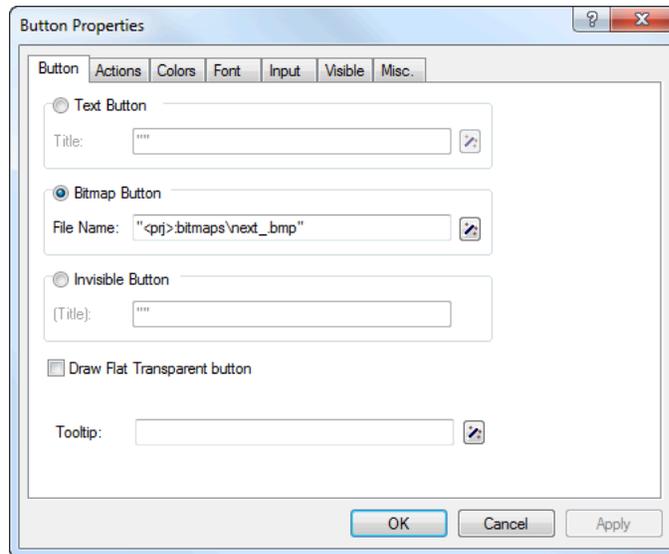
Figure 2.8: Bitmap button referencing a project user file

name (such as the **File Name** property of the bitmap button illustrated in Figure 2.8), you can easily select a project user file by pressing the wizard button , and selecting the **Select Project File** menu item. This will pop up a project user file selection dialog box similar to the dialog box shown in Figure 2.7.

2.6   Aimms licensing

Aimms offers the following two types of licenses:

Aimms *licensing*

- single-user licenses, and
- network licenses.

Each of these two types of licenses are protected in a different manner.

Single-user licenses can be used by a single user on a single computer. To enforce the single-user character, AIMMS requires that single-user licenses be protected by a nodelock file, which must be activated to match the hardware characteristic of your computer.

Single-user license protection

Nodelock files are stored on the harddisk of your computer, and are, therefore, much less vulnerable to loss. Only if you computer is stolen, or in case of a harddisk crash, you must contact AIMMS before being able to activate your nodelock on a replacement computer. In addition, you need access to the internet to activate or deactivate a nodelock.

Nodelocks

If you have ordered an AIMMS network license, no license protection needs to be installed locally on your computer. Instead, you need the host name and port number of the server running the AIMMS network license server. For more information about installing the network license server itself, please refer to the documentation of the AIMMS network license server.

Network licenses

2.6.1 Personal and machine nodelocks

AIMMS offers two types of nodelocks:

Two types of nodelocks

- personal nodelocks, and
- machine nodelocks.

If you choose for nodelock protection you are free to choose between a personal or a machine type of nodelock. In this section you will find the characteristics of both types of nodelocks. If you are unsure which type of nodelock to choose, we recommend that you start with a personal nodelock, as you can change a personal nodelock into a machine nodelock at any time, but not the other way around.

Personal nodelocks are intended for use by a single AIMMS user, who still wishes to have the freedom to use AIMMS on multiple computers, for instance if you want to easily switch between your desktop computer at work, a notebook computer and your home computer. Personal nodelocks have the following characteristics:

Personal nodelock

- Personal nodelocks can be transferred to another computer 3 times per 24 hours. This allows you to take your AIMMS license home in the evening and back to work the next morning without any problems.
- Personal nodelocks have a limited lifetime of 60 days, and should be renewed within that period to extend the lifetime to its full 60-day period. If the nodelock is not renewed within its 60-day lifetime, this does not invalidate your AIMMS license in any way—you only have to renew your nodelock prior to being able to use your AIMMS system again. Note that the renewal process does not require any manual intervention, as

AIMMS will try to automatically connect to our internet license database to renew your nodelock once every day you are using AIMMS.

- Both activation and nodelock renewal of personal nodelocks require a working connection to the internet. As a consequence, in the absence of an internet connection you can continue to work uninterrupted for a period of 60 days, before an internet connection is required to renew your nodelock.
- With every activation or nodelock renewal AIMMS will also update your license files if new license files are available (e.g. if your system is in maintenance), and will inform you of any messages that are available for you in our database.
- Because of their volatile nature, AIMMS will replace a personal nodelock without any questions asked in case of loss of or damage to your computer.
- You can switch your personal nodelock to a machine nodelock at any time.

Machine nodelocks are intended for permanent use on a single computer. They are recommended for server applications, and can also be used for personal use if you are sure you will be using AIMMS on a single computer, or do not have internet access. Machine nodelocks have the following characteristics: *Machine nodelock*

- Machine nodelocks can be transferred to a replacement computer 3 times per 365 days.
- Machine nodelocks have an unlimited lifetime (unless deactivated).
- Machine nodelocks can be either activated online if your computer is connected to the internet, or offline through the license activation area on the AIMMS website.
- License files will only be retrieved when the machine nodelock is activated, or by explicit request.
- In case of failure, AIMMS will, in principle, only replace machine nodelocks on the same computer.
- Once you have chosen for a machine nodelock, it is not possible to switch back to a personal nodelock.

Although a personal nodelock and the software version check on the start page make a regular connection to the internet (the personal nodelock connects to a license database for nodelock renewal and the version check connects to a version database), we do respect your privacy and will not register patterns in your personal usage of the AIMMS software in any way. During activation no personal information will be transferred, only your computer name and some of its hardware characteristics. During deactivation we register the date and time of deactivation to enforce the transfer limit. *Privacy*

The connection to our internet license database is implemented as a web service. Thus, if you are able to browse the web, you should also have no trouble activating an AIMMS nodelock. If your computer connects to the internet through a proxy server, AIMMS by default tries to detect and use the proxy settings also used by Microsoft Internet Explorer.

Internet connection and proxy settings

It should be noted that the use of auto-configuration scripts in determining the proxy server will fail if these use any other scripting language than Javascript. This is due to the libraries underlying the SOAP library used by AIMMS to connect to our license server. If you are in this situation, you should manually configure the proxy settings, as described below.

Automatic configuration scripts

If AIMMS does not succeed in automatically detecting the proxy settings that apply in your network environment, AIMMS also allows you to manually set the proxy settings during the activation process. If the online activation process does not succeed directly, AIMMS gives you the option to either continue with an offline activation process, or to manually supply the proxy settings that apply to your network environment through the dialog box illustrated in Figure 2.9. In this dialog box you can choose between

Manual proxy setting

- the *Current User* settings also used by Microsoft Internet Explorer (default),
- the `Local Machine` settings which are stored in the registry, if these are available on your machine, or
- *Custom* proxy settings that you have received from your IT department.

In the latter case, you can also (optionally) provide a user name and password to authenticate with the proxy server. In most cases, however, setting these will not be necessary, and Windows authentication will be sufficient.
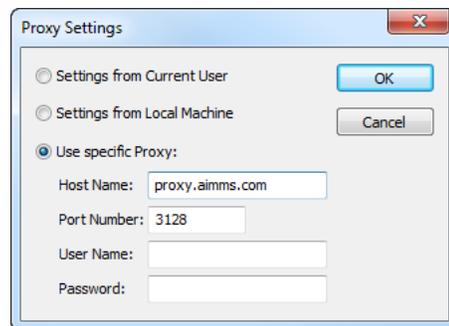

Figure 2.9: The AIMMS **Proxy Configuration** dialog box

2.6.2 Installing an Aimms license

When you start up Aimms for the first time after installation, Aimms will open the **License Configuration** dialog box illustrated in Figure 2.10. Through this

Managing your Aimms licenses
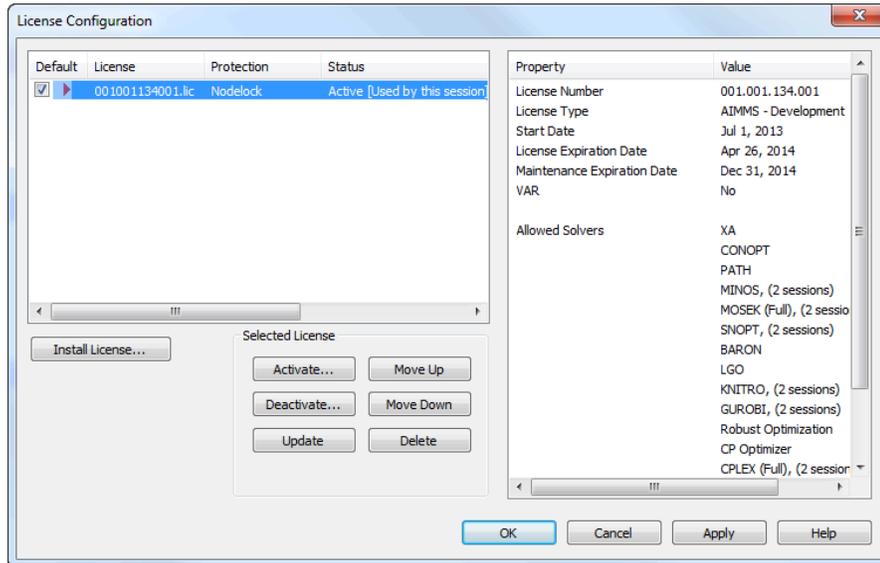
Figure 2.10: The **License Configuration** dialog box

dialog box you can install new Aimms licenses and manage all Aimms licenses that already have been installed on your computer.

To install a new license, press the **Install License ...** button in the **License Configuration** dialog box. This will start a wizard, that will guide you through the license installation procedure step by step. The wizard can help you to install

Installing a new license

- existing Aimms licenses,
- nodelocked licenses,
- network licenses,
- evaluation licenses, and
- student licenses.

To successfully complete the installation of licenses of each type, you should make sure to have the following information available.

To install a single-user AIMMS license that is protected by a nodelock, you need the following information:

- your AIMMS license number, and
- the associated activation code that you received from AIMMS.

Single-user nodelocked licenses

You have the choice to request a personal nodelock or a machine nodelock. A personal nodelock must be requested online, a machine nodelock can be requested online or offline. Refer to Section 2.6.1 for a more detailed introduction to personal and machine nodelocks.

To install an AIMMS network license, you need the following information from your system administrator:

- the name of the AIMMS network license server,
- the port number of the AIMMS network license server, and
- the name of the license profile to which you want to connect (optional).

Network licenses

To install an AIMMS evaluation license you need the following information

- your AIMMS evaluation license number, and
- the associated activation code that you received from AIMMS when requesting an evaluation license.

Evaluation licenses

You must have a working connection to the internet (not necessarily on the machine on which you installed AIMMS) to activate an evaluation license. Evaluation licenses expire 30 days after activation. Note that each evaluation license can be activated only once, and that you can only activate a single evaluation license per AIMMS release on a specific computer, regardless of the number of evaluation licenses you have requested on our web site.

To install an AIMMS student license you need the following information:

- your AIMMS student license number, and
- the associated activation code that you received from the university that purchased the AIMMS Educational Package.

Student licenses

You must have a working connection to the internet to activate a student license. Student licenses expire one month after the end of the current academic year. Student licenses can be activated multiple times.

2.6.3 Managing AIMMS licenses

AIMMS allows you to have multiple AIMMS licenses installed on your computer. You may have multiple licenses installed, for instance, for the following reasons:

Managing multiple AIMMS licenses

- you have requested a trial license for a new AIMMS version which you want to run next to your existing license,
- you have temporarily borrowed or hired an AIMMS license with more capabilities than your regular license,
- your system administrator has created multiple network license profiles, each of which you may want to use to run AIMMS.

In this section we will describe how you can instruct AIMMS which license to use.

In the **License Configuration** dialog box displayed in Figure 2.10, all AIMMS licenses installed on your machine will be displayed in the left pane of the dialog box. The license details of each license are displayed in the right pane of the dialog box. During startup AIMMS will consider all licenses in the left pane of the **License Configuration** dialog box which have the **Default** column checked, and will use the first valid license it finds starting from top to bottom. Using the **Move Up** and **Move Down** buttons you can change the order in which AIMMS will search the list.

Default licenses

Both personal and machine nodelocks can be transferred to other computers. Personal nodelocks can be transferred upto three times a day, allowing you to take your license with you wherever you want. Machine nodelocks can be transferred three times per year, to a computer replacing the computer on which the nodelock is currently installed. To transfer a nodelocked license, you must

Transferring licenses

- deactivate the nodelock on the currently active computer, and
- activate it on the computer to which you want to transfer the license.

You can deactivate an active nodelock using the **Deactivate** button in the **License Configuration** dialog box. Deactivation will only succeed if there is no conflict with the transfer limit for the given nodelock type. This makes sure that there will never be a problem activating a deactivated license. After successful deactivation the license will not be removed from the list but be marked as inactive. If the license is not active on any computer, you can reactivate the license through the **Activate** button.

In case you want to activate a nodelock on a computer, but have forgotten to deactivate the nodelock on a computer to which you currently have no access, AIMMS allows you, as a courtesy, to request an emergency nodelock 3 times per 365 days. Emergency nodelocks have a lifetime of 7 days, and during this time you can arrange for someone to deactivate the license on the computer containing the active nodelock. During the activation sequence, AIMMS will automatically ask whether you would like to receive an emergency nodelock when it discovers that the license is active on another computer.

Emergency nodelocks

2.6.4 Location of license files

Aimms keeps its license and configuration files in the folder

    `AIMMS`

Location of license files

of the common application area of your computer. On Windows 7, 8 and 10, this folder is located under `C:\ProgramData`. The `AIMMS` folder contains five sub-folders

- `Config`, containing the license and solver configuration files,
- `Licenses`, containing all license files,
- `Nodelocks`, containing all nodelock files installed on your computer,
- `ApplicationKeys`, containing any private/public key pairs that you have generated, and
- `AnyUser`, containing the license configuration files for all users on your computer (see below).

The Aimms installation makes sure that these subfolders are writable for everyone, allowing you to install and uninstall licenses on your computer.

To prevent tampering with nodelocked licenses, Aimms keeps track of the location of the nodelock files associated with a license. You should, therefore, not manually move or copy the Aimms nodelock files as this may invalidate your nodelock.

Do not move nodelock files

You can specify whether the license and solver configuration that Aimms uses is the same for any user of the machine, or different for each individual user. To prevent problems when running Aimms as part of a computer service, Aimms will by default use the same configuration for any user. To modify this behavior, you should edit the file `UserDistinction.cfg` in the common `Config` folder. In the file `UserDistinction.cfg.default`, straightforward directions are given on how to edit it. If no `UserDistinction.cfg` file exists, Aimms will use the `UserDistinction.cfg.default` file instead. You can use this file as a base for setting up your own configuration.

User specific configuration

In the scenario where all users of the same pc use the same license and solver configuration, the configurations that are modified by a user are stored in the `AnyUser` folder of the `Common` folder.

Any User

In the scenario where each specific user of the pc has its own license and solver configuration, the configurations that are modified by a user are stored in the local application data folder. On Windows 7, 8 and 10 it is:

Current User

    `C:\Users\<UserName>\AppData\Roaming\AIMMS`

If you have already used AIMMS 3 on your machine, these folders don't end in AIMMS, but (still) in `Paragon Decision Technology`.

When AIMMS needs to read the current configuration, it will first look in the (any)user folder as specified by the aforementioned `UserDistinction.cfg` file; if it cannot be found there, it will try to read the configuration from the common application data folder. When saving a modified configuration, AIMMS always writes to the (any)user folder.

Accessing configuration files

Inside the `Config` folder of the (any)user folder, each major AIMMS version (3.6, 3.7, 3.8, ..., 4.0 etc.) will create its own specific subfolder when it needs to write a configuration file. During an attempt to read, AIMMS will first look for the specific file in the folder that matches its own major version number, and otherwise it will subsequently search through the folders of previous versions. In other words, when you upgrade to a new AIMMS version, initially your configuration will be the same as the one you were using for the previous AIMMS version, but if you change something in the configuration, this will only affect the configuration of the AIMMS version you are working with.

Version dependent configurations

If you are using an AIMMS Network License, then your local machine does not need to have any license file installed. The only required file is the license configuration file, that contains the info of where the License Server is located on your LAN. When logging on to the License Server, the licensing info is sent directly from the server to the running AIMMS session, except for some secondary license related files:

Network License Client Files

- the .SLV file (containing the default solver configuration), and
- the .CPX file (the *CPLEX* license file).

These secondary license files are temporarily copied to the folder `NetworkCache` which is located in the (any)user folder as described above.

You can specify a project dependent solver configuration by placing a solver configuration file with the name 'solvers.slv' in the project directory. AIMMS will first look for this file and if it cannot find it will look for other solver configuration files. See the AIMMS Help for more information.

Project dependent configuration